

Génie Logiciel

Niveau : 3^{ème} année licence (MI)

Chapitre 5 : Diagrammes UML (vue dynamique - Partie 2)

Chargés de cours :

Dr M. MOHAMMEDI
& Dr N. YESSAD

Année universitaire 2023/2024

Plan du chapitre 5 - Partie 2



Diagramme d'états de
transitions



Diagramme d'activités

Diagramme d'états-transitions

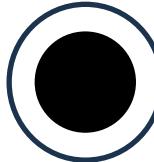
- Le diagramme d'états-transitions décrit le comportement **dynamique** d'une entité (Logiciel, composant, objet ...)
- Il représente les différents **états** (situations) dans lesquels peut se trouver l'entité ainsi que la façon dont cette entité **passe d'un état à l'autre** en **réponse à des événements**.
- Un objet peut passer par une série d'états pendant sa durée de vie.
- Il repose sur différents concepts :
 - les états
 - les transitions et les événements
 - les actions et les activités.

Concepts de base: Etat

- Un état est une **situation stable** qui possède une **certaine durée** pendant laquelle **un objet exécute** une **activité** ou attend **un événement**.
- Un état est une abstraction d'un moment de la vie d'une entité pendant lequel elle satisfait un ensemble de conditions
- Un état est caractérisé par: les valeurs des rubriques de l'objet et l'existence des associations/liens de cet objet aux autres objets.
- Un état est représenté par un **rectangle arrondi** contenant son nom.

ETAT 1

Concepts de base: Etat

- L'état **initial** est l'état dans lequel un objet se trouve lors de sa création
- Il est représenté par un point noir 
- Il est obligatoire et unique
- Un état **final** correspond à une étape où l'objet n'est plus nécessaire dans le système et où il est détruit. L'état final se représente par un point entouré d'un cercle.
- Il est optionnel, peut être multiple

Concepts de base: Evènement

- Un évènement est une **information instantanée** venant de l'extérieur du système, survenant à un **instant donné** qui doit être traitée à l'instant où il se produit.
- Un événement est un fait **qui déclenche le changement d'état**, qui fait donc **passer** un objet **d'un état à un autre état** (désactivation d'un état et activation de l'état suivant).
- Quand un événement est reçu, une **transition** peut être déclenchée et faire basculer l'objet dans un nouvel état
- Syntaxe d'un événement :
 - Nom de l'événement (Nom de paramètre : Type,...)
 - La description complète d'un événement est donnée par : Nom de l'événement, Liste des paramètres, Objet expéditeur, Objet destinataire, Description textuelle

Concepts de base: Type d'évènement

Événement de type signal:

Correspond à la réception **d'un signal asynchrone** émit par un autre objet ou par un acteur;

Événement appel d'opération:

représente la réception **de l'appel d'une méthode** de l'objet courant par un autre objet ou par un acteur

Événement de changement:

généré par **la satisfaction d'une expression booléenne** sur des valeurs d'attributs « when (<condition_booléenne>) »

Événement temporel:

généré par **l'écoulement du temps**, soit de manière absolue (date précise «when [date=<date>»]), soit de manière relative (temps écoulé « after<durée> »)

Concepts de base: Transition

- Une transition est **le passage d'un état** (état source) à **un autre état** (état cible).
- Une transition est déclenchée: **soit par un événement, soit automatiquement** lorsque aucun événement déclencheur est spécifié.
- Une transition est représentée **par une flèche orientée** de l'état source vers l'état cible. Elle est **étiquetée par l'événement** qui provoque le changement d'état.



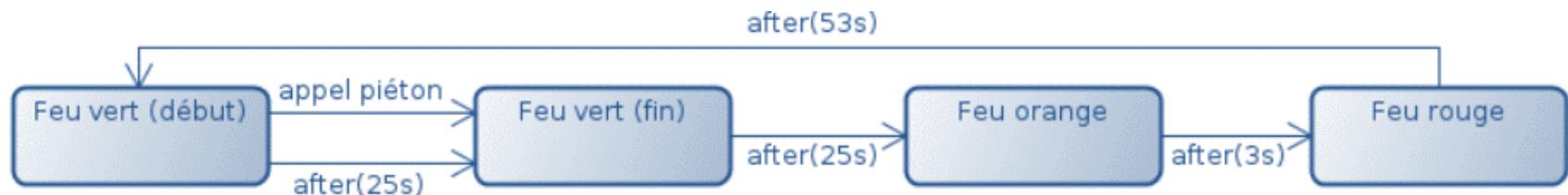
Concepts de base: Transition

□ Transition externe:

-Définit **la réponse** d'un objet à l'arrivée d'un événement.

-Elle indique qu'un objet qui se trouve dans un état peut « **transiter** » vers un autre état en exécutant éventuellement certaines activités, si un événement déclencheur se produit et qu'une condition de garde est vérifiée

-Est une transition **qui modifie l'état actif**.



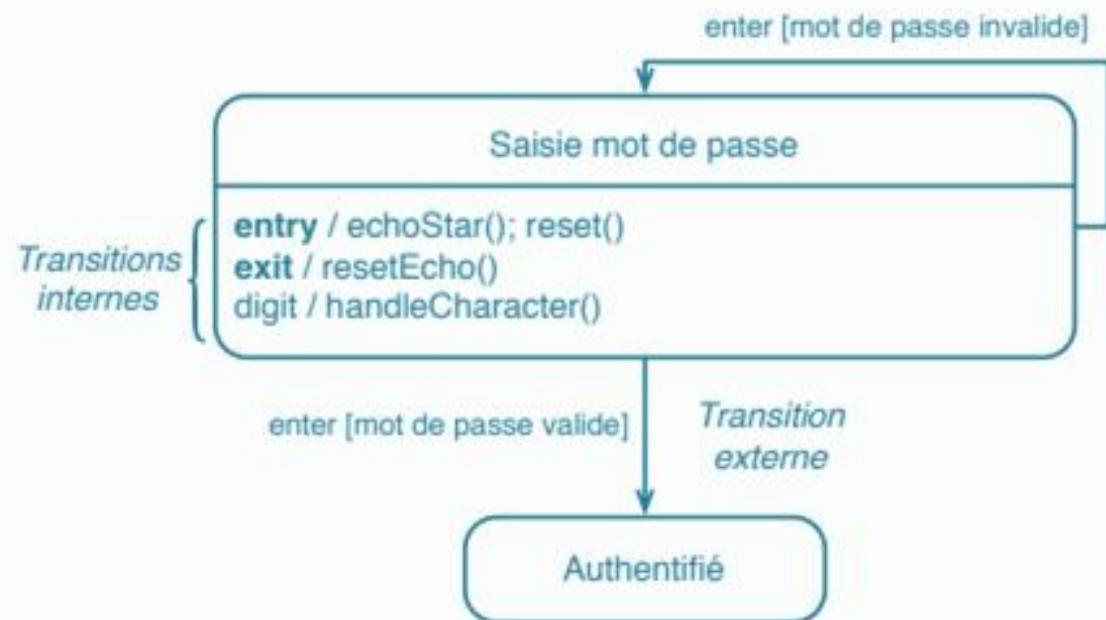
Concepts de base: Transition

□ Transition interne

- Une transition interne **ne possède pas d'état cible** et que l'état actif **reste le même** à la suite de son déclenchement (n'engendre pas de changement d'état).
- N'est pas représentée par des arcs mais est spécifiée dans un **compartiment de son état associé** (rattachée à un état).
- Les transitions internes possèdent des noms d'événement prédéfinis correspondant à des déclencheurs particuliers : **entry, exit, do et on**.
 - Ces mots clefs réservés viennent prendre la place du nom de l'événement dans la syntaxe d'une transition interne.

Concepts de base: Transition

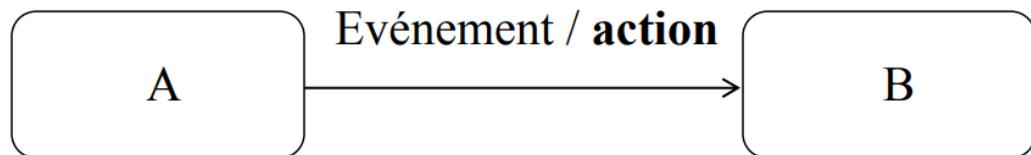
- **entry** : définit l'activité à exécuter lors de l'entrée dans l'état.
- **exit** : définit l'activité à exécuter lors de la sortie de l'état.
- **do** : définit l'activité à exécuter dès que celle définie par entry est terminée.
- **on** : (optionnel) définit l'activité à exécuter à chaque fois que nous avons un événement particulier.



Concepts de base: Actions/Activités

Actions

- Une action est une opération **instantanée et atomique**
- Une action **est associée à un événement** (accompagne un changement d'état)
- Elle a accès aux paramètres de l'événement et aux attributs de l'objet.
- Une action interne est utilisée lorsqu'un événement déclenche une action sans changement d'état



- Types d'actions

- Affectation
- Appel / envoi de message
- Création d'instance
- Destruction d'instance
- Séquence
- Divers: [description] (Décrise dans une autre langage)

Concepts de base: Actions/Activités

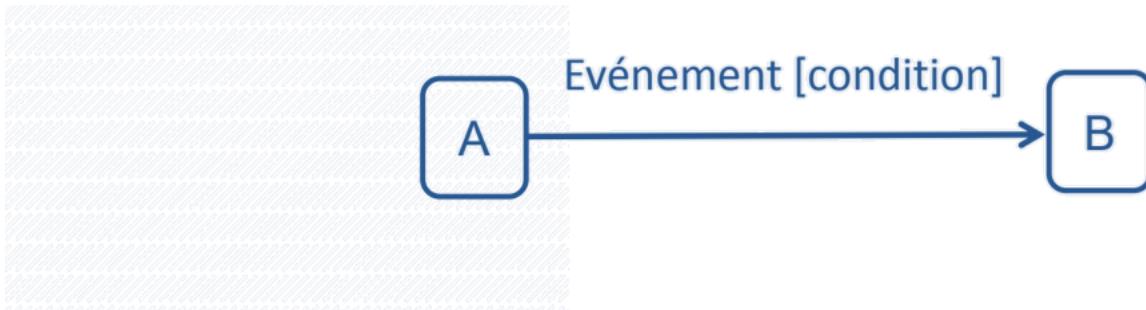
Activités

- Une activité est **une série d'actions**
- Une activité représente une opération qui nécessite un certain temps d'exécution.
- Une activité **est non atomique**. Elle peut être **interrompue à tout moment** par un événement générant une transition.
- Une activité prend un temps non négligeable et est exécutée pendant que l'objet est dans un état donné.
- Une activité est associée à un état mais un état peut ne pas avoir d'activité.

Remarque: Actions et activités sont exprimés par des verbes.

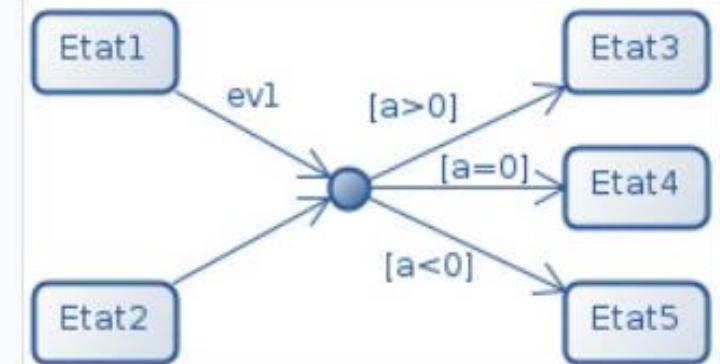
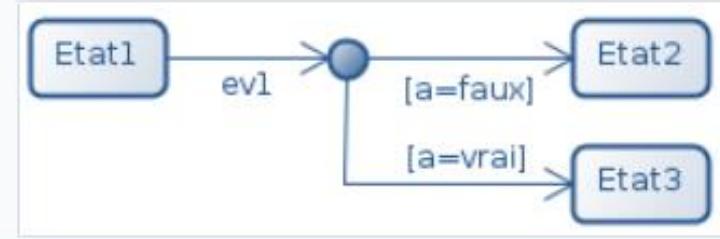
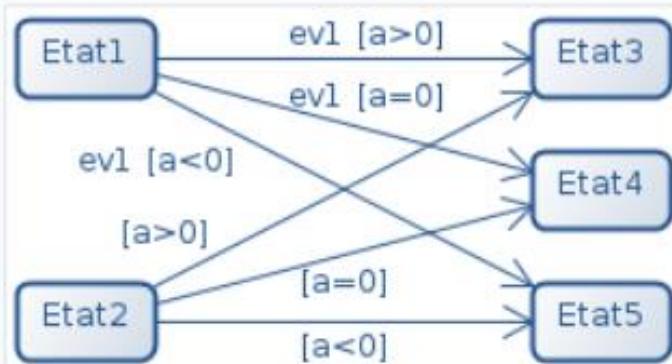
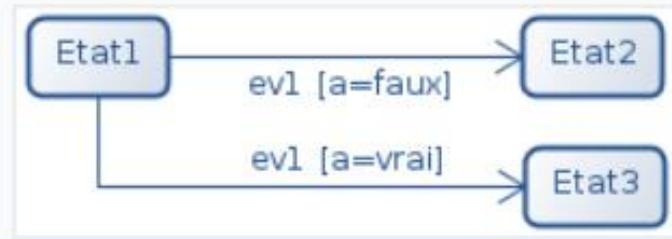
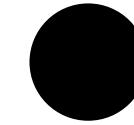
Notion de gardes

- Il est possible d'exprimer des **conditions booléennes** dont dépend le déclenchement d'une transition lors de l'occurrence d'un événement.
- Ces conditions sont appelées « **gardes** ».
- Les gardes sont notées **entre crochets** (spécifiée par '[]' dans la syntaxe).
- Il s'agit d'une **expression logique** sur les attributs de l'objet
- Évaluées uniquement quand l'événement est déclenché
 - Si la garde est vraie, la transition est réalisée
 - Si la garde est fausse, la transition n'a pas lieu
- Lorsqu'un même événement est associé à plusieurs transitions, une garde peut déterminer la transition à effectuer



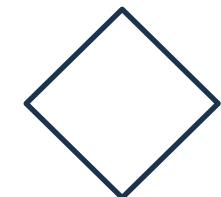
Point de jonction

- Les points de jonction permettent à **plusieurs transitions d'avoir une partie commune** en partageant des segments de transition.
- L'utilisation de points de jonction a pour but de rendre la notation des transitions alternatives plus lisible.
- Les points de jonction sont représentés par **un cercle plein** :



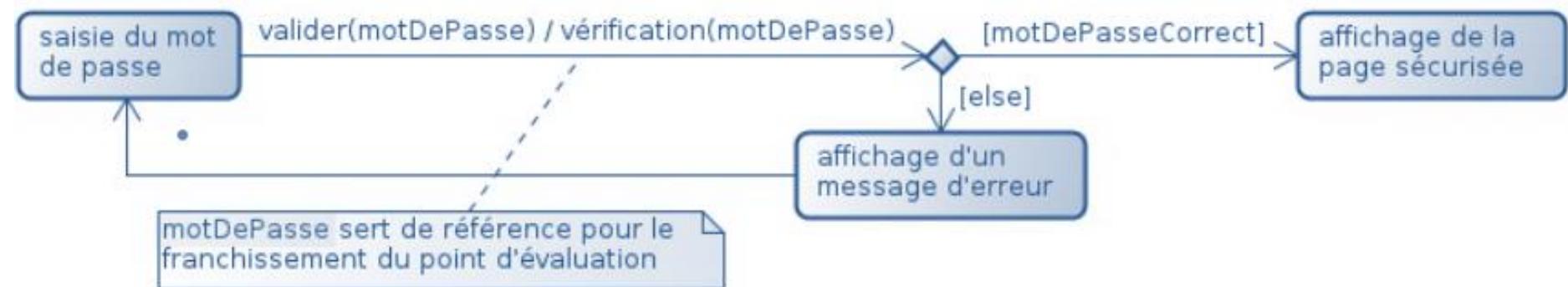
Point de décision

- Les points de décision ont un fonctionnement similaire à celui du point de jonction mise à part que nous pouvons sortir de l'état source dès que le segment avant le point de décision est franchissable.
- Le choix du segment à franchir derrière le point de décision se fait au moment de l'arrivée sur **le point de décision**. Cela permet aux segments qui sont derrières le point de décision d'avoir une condition de garde qui dépend d'éléments qui sont définis par une activité effectuée lors du franchissement du segment de transition avant le point de décision.
- Les points de décision sont **représentés par des losanges** :



Point de décision

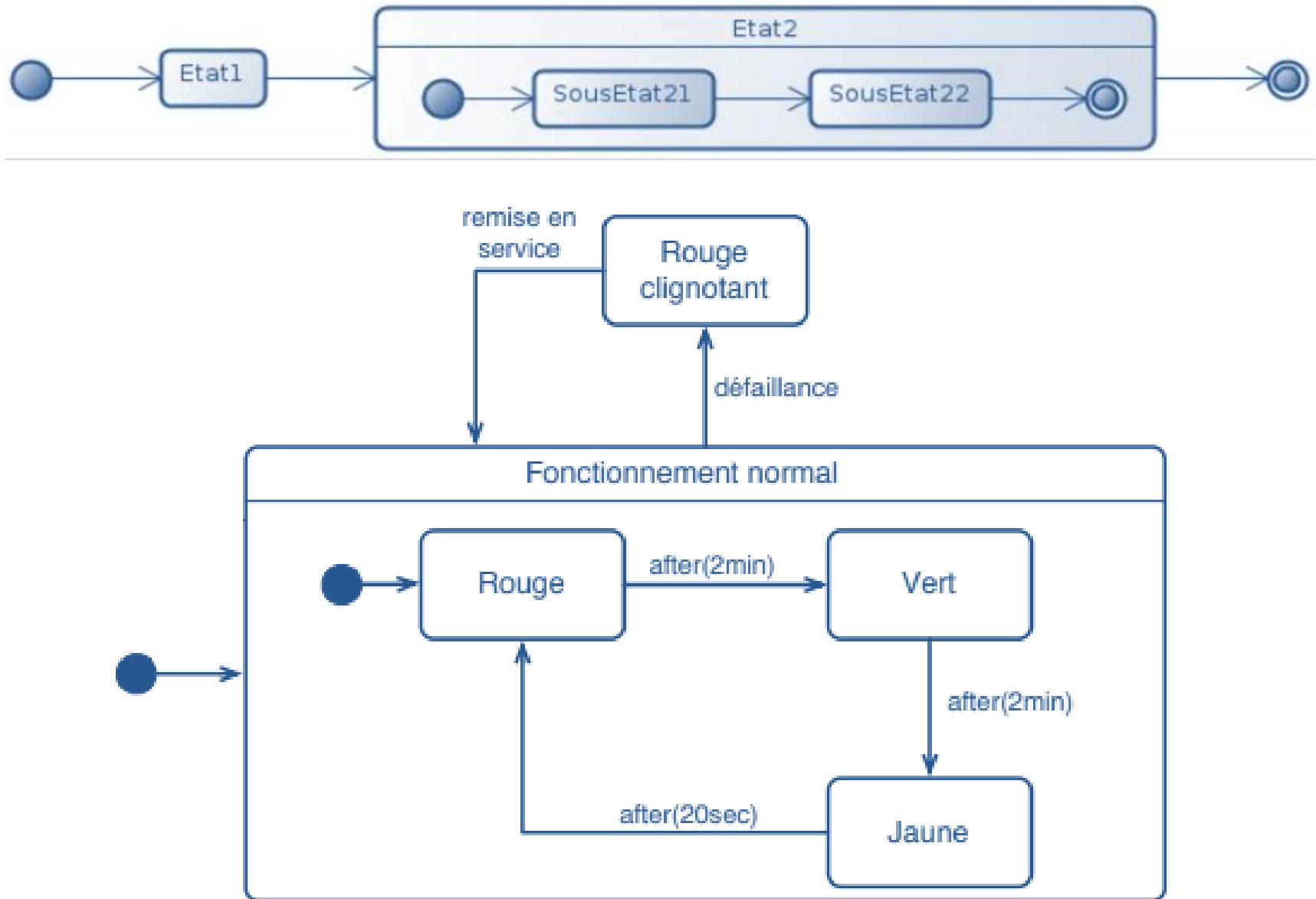
Remarque: Lorsque nous arrivons sur un point de décision, il faut qu'un des segments de transition qui suit, soit franchissable (une transition n'ayant pas de durée, nous ne pouvons pas rester à attendre quelque chose au niveau du point de décision). Pour éviter ce problème nous pouvons ajouter un segment avec la garde **[else]** qui est automatiquement franchie si aucune des gardes des autres segments n'est vraie. Un segment de transition derrière un point de décision ne peut pas comporter d'événement.



Etats composites

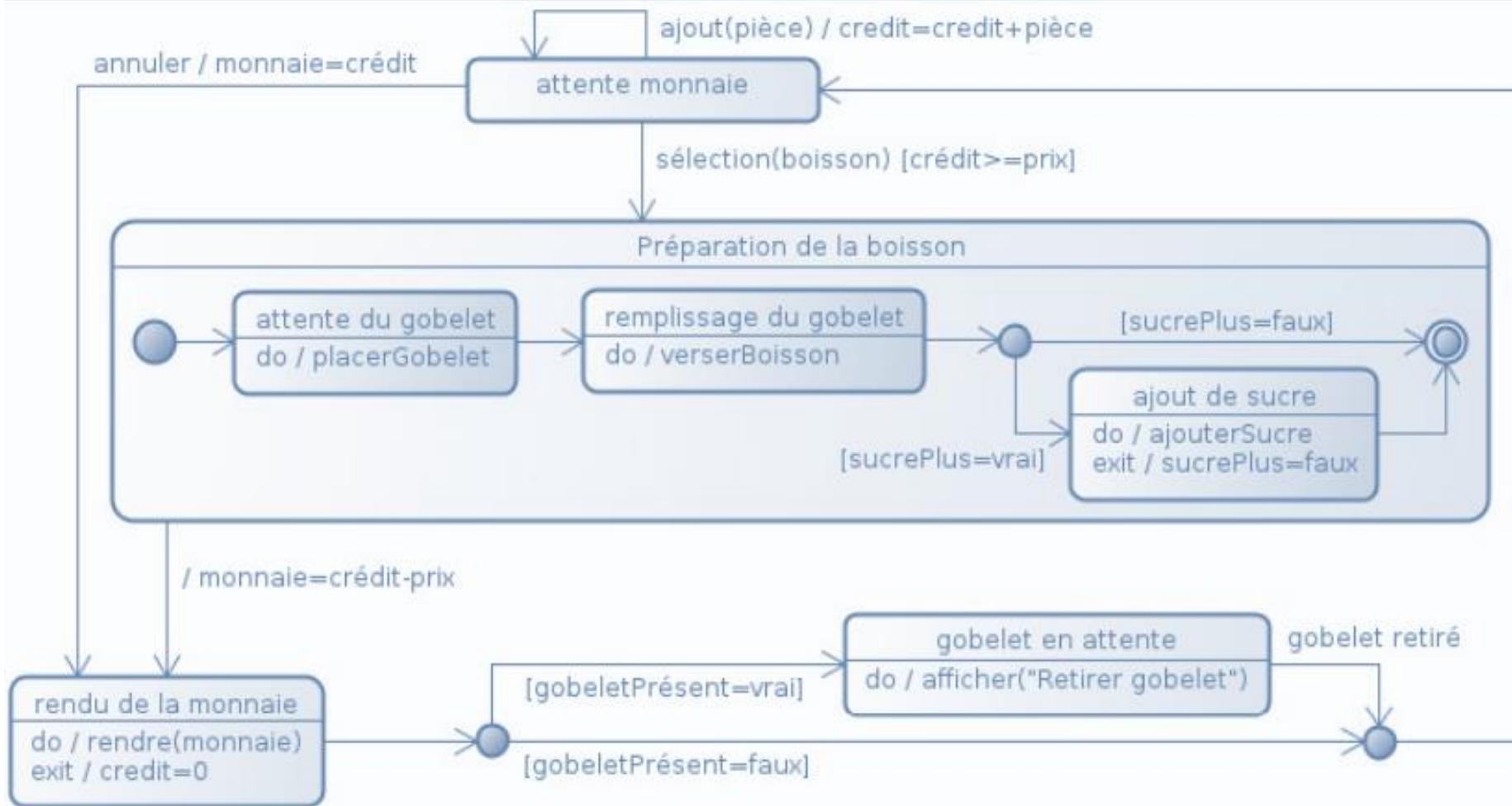
- Certains états sont complexes et correspondent à la réalisation de plusieurs activités qui ne pourront pas être définis par des transitions internes.
- Il peut alors être intéressant de le **décomposer en sous-états**.
- Un état composite est un **état décomposable en sous-états**.
- Un état peut être partitionné **en plusieurs compartiments séparés par une ligne horizontale**. Le premier compartiment contient le nom de l'état et les autres peuvent recevoir des transitions internes, ou des sous-états, quand il **s'agit d'un état composite**.
- Dès qu'un état composite est actif, il active son sous état initial.
 - Si un état composite est raccordé à une transition automatique, elle est franchie lorsque nous atteignions le sous état final de l'état composite.
 - Si une des transitions externes attenantes à l'état composite est franchissable alors elle est franchie et tous les sous-états deviennent inactifs.

Etat composites



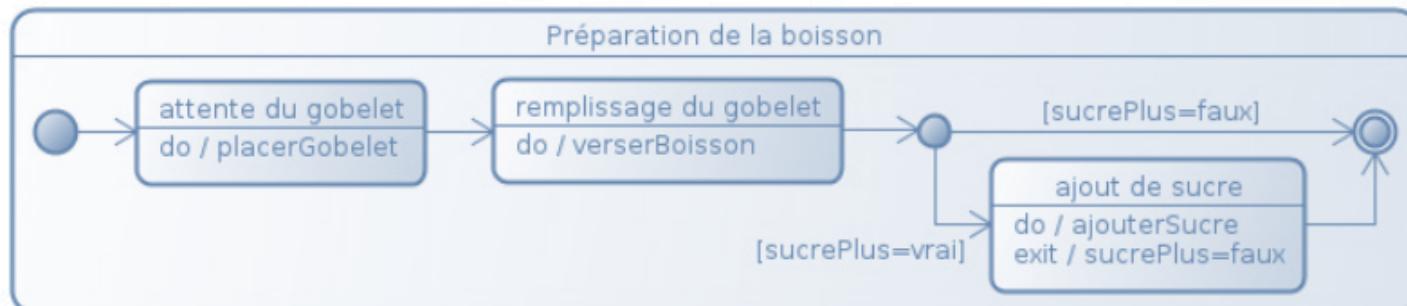
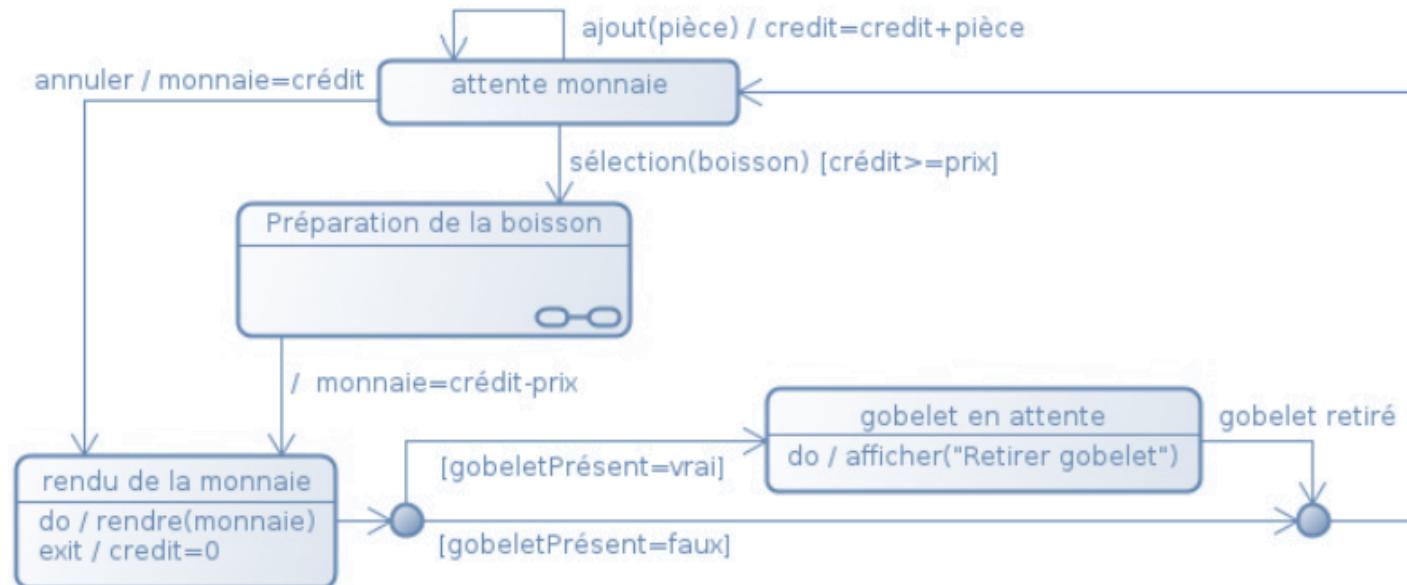
Etat composites

Exemple : le distributeur automatique de boisson



Etat composites

Afin d'éviter de surcharger le diagramme d'états-transition, il est possible de **placer le symbole (o-o)** à l'intérieur de l'état composite et de spécifier ensuite son contenu ailleurs dans une autre page.

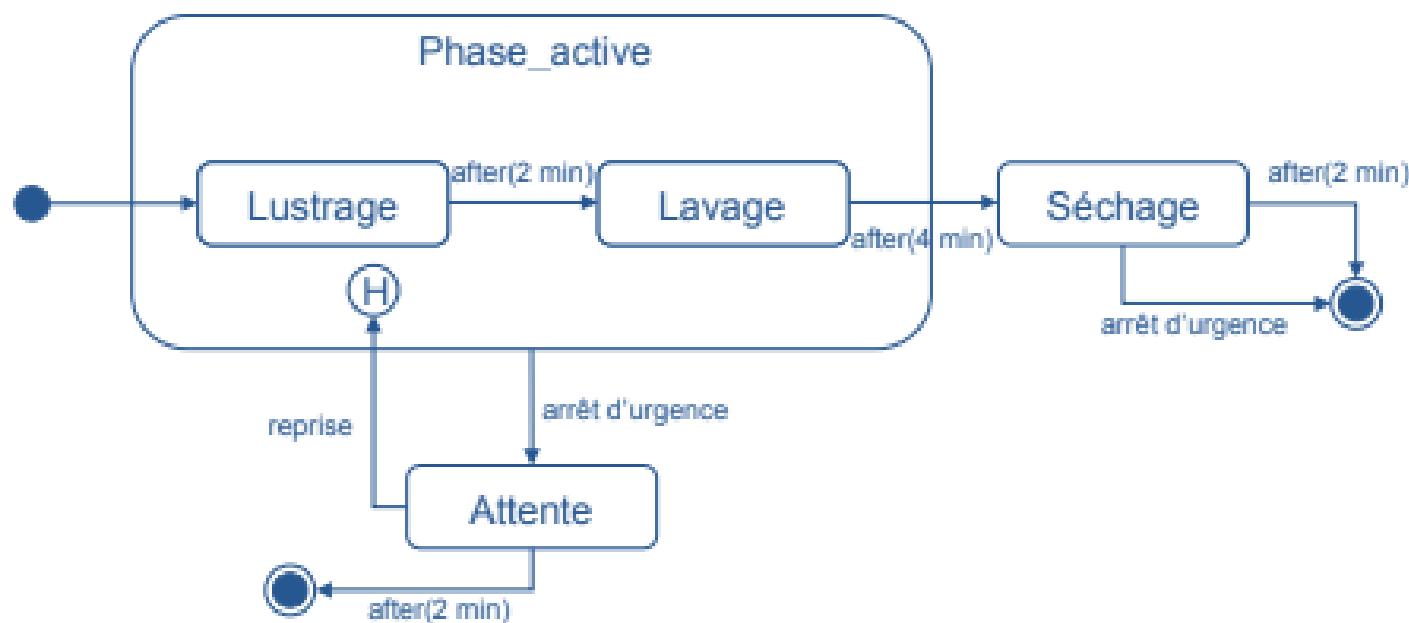


Etat historique

- Un état à historique permet à un état composite qui contient des sous-états **de se souvenir du dernier sous-état** avant la transition réalisée depuis l'état composite.
- Permet de restaurer l'état suite à une sortie temporaire d'un état composite
- Un état historique est **désigné par**  ou 
-  pour reprendre au début du sous-état du plus haut niveau dans lequel nous nous étions arrêté.
-  pour reprendre au début du sous état dans lequel nous nous étions arrêté, quelque soit son niveau d'imbrication (historique profond).

Etat historique

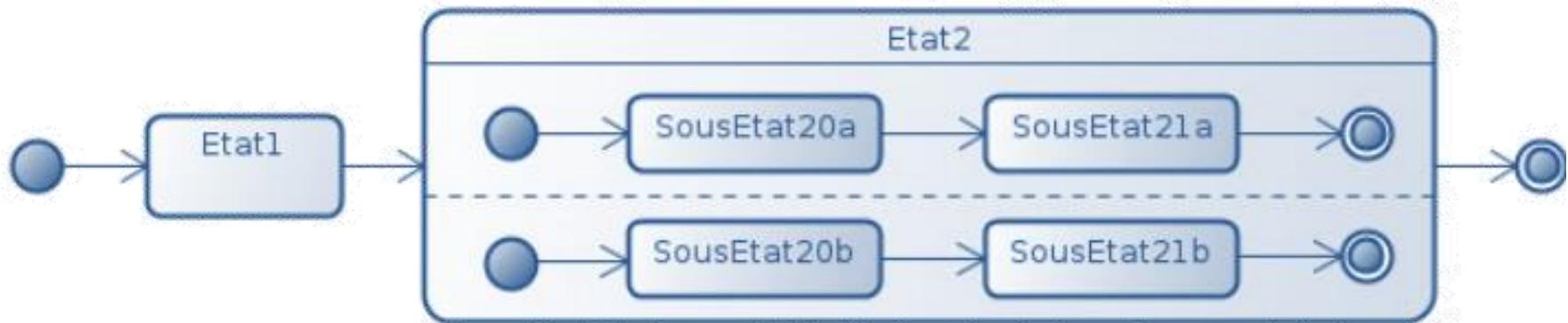
- Exemple : une machine à laver peut être arrêtée dans un état(lavage, rinçage, essorage). Elle redémarrera du même état.



Etat orthogonal / Concurrence

- Les diagrammes états-transitions permettent de décrire efficacement les mécanismes **concurrents** grâce à l'utilisation d'états orthogonaux.
- Un état orthogonal est un état qui possède **plusieurs régions (séparées par des pointillés horizontaux)** qui évoluent **simultanément et en parallèles**;
- Chaque région représente un flot d'exécution, elle peut posséder un état initial et un état final.
- Une transition qui atteint la bordure d'un état composite orthogonal est équivalente à une transition qui atteint les états initiaux de toutes ses régions concurrentes.
- Toutes les régions concurrentes d'un état composite orthogonal doivent attendre leur état final pour que l'état composite soit considéré comme terminé.

Etat orthogonal / Concurrence

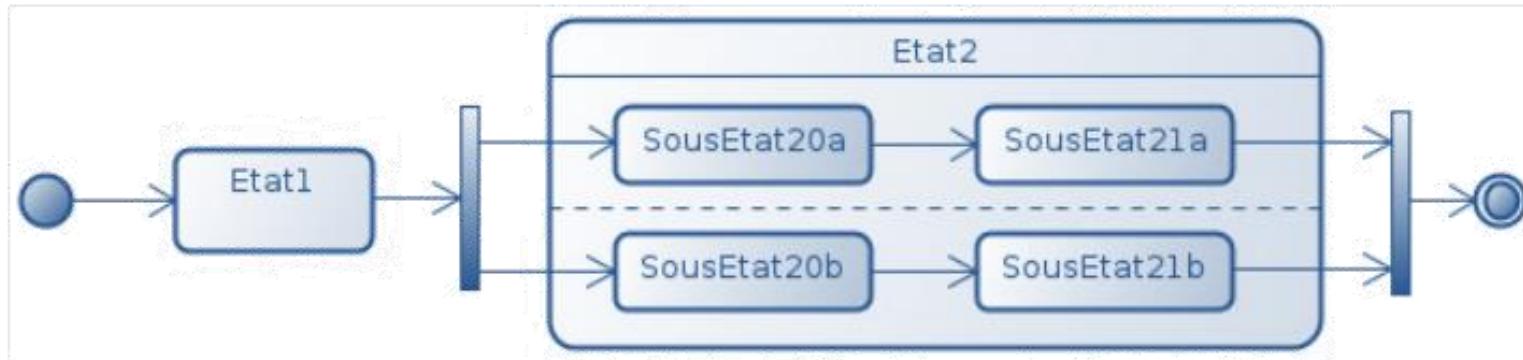


Lorsque l'état2 est activé, l'activation des états initiaux de chaque région est exécutée.

- Si la transition externe qui permet de sortir de l'état2 est une transition automatique, elle est franchie uniquement lorsque toutes les régions ont leur état final actif.

Etat orthogonal / Concurrence

Le tracé précédent est très clair, mais il est aussi possible de représenter des flots d'exécutions parallèles en utilisant la notation des transitions concurrentes (barres épaisses) :



Exercice

On considère une boîte de vitesses automatique de voiture. La boîte au démarrage est au point mort. La marche arrière ainsi que la position parking peuvent être enclenchées à partir du point mort. La première marche avant peut également être enclenchée à partir du point mort. En revanche, les autres marches avant, la seconde et la troisième, sont enclenchées en séquence: $1 \rightarrow 2 \rightarrow 3$ pour une accélération, et $3 \rightarrow 2 \rightarrow 1$ pour une décélération. Seules la marche arrière, la position parking et la première marche avant peuvent être ramenées directement au point mort.

Fin du Chapitre 5.

Partie 2

